

Docket No.: 62758-072

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of	:	Customer Number: 20277
	:	
Ryoichi UEDA, et al.	:	Confirmation Number:
	:	
Serial No.:	:	Group Art Unit:
	:	
Filed: February 25, 2004	:	Examiner: Unknown
	:	
For:		DOCUMENT STRUCTURE INSPECTION METHOD AND APPARATUS

**CLAIM OF PRIORITY AND  
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Mail Stop CPD  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

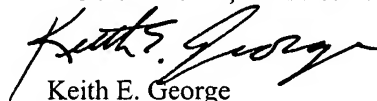
In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

**Japanese Patent Application No. 2003-140568, filed May 19, 2003**

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY



Keith E. George  
Registration No. 34,111

600 13<sup>th</sup> Street, N.W.  
Washington, DC 20005-3096  
(202) 756-8000 KEG:tlb  
Facsimile: (202) 756-8087  
**Date: February 25, 2004**

NT1460 (24) 7,100  
Q2758-072  
UEDA, et al.

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

February 25, 2004

*McDermott, Will & Emery*

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    5 月 1 9 日  
Date of Application:

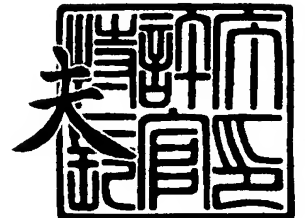
出 願 番 号                      特 願 2 0 0 3 - 1 4 0 5 6 8  
Application Number:  
[ST. 10/C] :                      [ J P 2 0 0 3 - 1 4 0 5 6 8 ]

出      願      人                      株式会社日立製作所  
Applicant(s):

2 0 0 4 年    1 月 2 8 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号    出証特 2 0 0 4 - 3 0 0 3 6 8 7

【書類名】 特許願

【整理番号】 NT03P0254

【提出日】 平成15年 5月19日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/60

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 植田 良一

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 中山 弘二郎

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

## 【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

## 【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 文書構造検査方法および装置

【特許請求の範囲】

【請求項 1】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、事前に取り決めた文書構造変化規則に基づき文書構造定義を変換し、変換後の前記文書構造定義に基づき前記構造化文書の構造を検査することにより、前記文書構造が変化した前記構造化文書の、前記文書構造変化の検査者による可逆性に依存しない検査を実現することを特徴とする文書構造検査方法。

【請求項 2】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造を検査中にエラーが発生した場合、文書構造変化規則集から適用可能な文書構造変化規則を検索し、検査に利用する文書構造定義を、検索した文書構造定義に切り替えて検査を続行することにより、前記文書構造が変化した前記構造化文書の、前記文書構造変化の検査者による可逆性に依存しない検査を実現することを特徴とする文書構造検査方法。

【請求項 3】

請求項 1 または 2 に記載の文書構造検査方法において、文書構造定義言語毎に文書構造検査部を準備し、文書構造定義の記述に利用している文書構造定義言語に応じて、文書構造検査部を入れ替えることにより、前記文書構造定義が複数の言語により記述されている場合でも検査可能とすることを特徴とする文書構造検査方法。

【請求項 4】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造変化規則集に記憶された文書構造変化規則に従って、文書構造定義集に記憶された文書構造定義を変換する手段と、該変換手段により変換された文書構造定義を利用して文書構造を検査する手段とを有することを特徴とする文書構造検査装置。

【請求項 5】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造を検査する手段と、前記文書構造の検査中にエラーが発生した場合文書構造変化規則集から適用可能な文書構造変化規則を検索する手段と、検査中に検査に利用する文書構造定義を切り替える手段とを有することを特徴とする文書構造検査装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、構造化文書の構造検査方法に関する。

【0002】

【従来の技術】

一般に通信ネットワークを介して処理要求を受け取り、要求に見合った処理を行う計算機システムでは、要求内容を表す文書（ネットワーク上で送受信される電子データを意味する）を受け取るたびにそのデータ構造が事前に取り決めた構造と合致するかどうかを検査し、文書の構造が正しい場合にのみ主となる業務処理を行い、その結果を返信する。文書の構造が正しくない場合はエラーを返信する。

【0003】

このような文書構造検査処理は業務処理を実行しながら徐々に実施することも可能であるが、業務処理の最終段階になって文書構造が正しくないことがわかった場合、それ以前の全ての業務処理が無駄になってしまい、処理効率が悪化する。このような処理効率の悪化を防ぐために、業務処理の前に文書構造全体の検査が行われることが望ましい。これは送受信される情報がXML（E x t e n s i b l e M a r k u p L a n g u a g e）で表現されている場合も同様である。

【0004】

個々の構造化文書の構造があらかじめ定義された文書構造定義と合致していることを検査する処理は、J a v a（登録商標）を対象として、R a j i v M o r d a n i 他が、「J a v a A P I f o r X M L P r o c e s s i n g

Version 1.2 Final Release」(非特許文献1)などで「Schema Validation」としてそのインターフェイスを規定しており、The Apache Software Foundation、「Xerces2 Java Parser Readme」(非特許文献2)などでは、既に実装されていることが述べられている。

#### 【0005】

また、前記のような通信ネットワークを介して情報を受け取る計算機システムでは、送信文書の改竄防止や盗聴防止などの目的で送信データに送信者の電子署名を付与したり、送信文書を暗号化するといった事が行われる場合がある。

#### 【0006】

さらに、複数の計算機が順番にそれぞれの処理を行い、全体としてより複雑な処理を実現する計算機システムの場合には、最初の送信者(処理要求者)は送信データの一部分に電子署名を付与したり、一部分を暗号化したりすることがある。

#### 【0007】

このような処理を行う例として、オンラインショッピングがある。オンラインショッピングでクレジットカードを利用して支払いを行う場合、購入者(最初の送信者)からオンラインショッピングサイト運営者に送信されるデータには、購入するものを特定するための情報と、支払いに利用するクレジットカードを特定するための情報(カード番号、有効期限など)が含まれる。このうちクレジットカードに関する情報がオンラインショッピング運営者からクレジットカード会社に送信され、実際のクレジット決済処理が行われる。オンラインショッピング運営者はクレジットカード情報を直接利用するわけではないが、購入者から受け取ってクレジットカード会社に中継する役割を担う。

#### 【0008】

このようなシステムでは、送信データ全体をオンラインショッピング運営者が復号化できる方式で送信する方式よりも、利用者側でクレジットカード会社だけがクレジットカード情報を復号化できる方式で暗号化(部分暗号)し、オンラインショッピングサイト運営者にはクレジットカード情報を開示しない方式の方が

安全性が高い。また、この時、クレジットカード情報が購入者自身により暗号化されたことを示すために暗号化後のデータに電子署名（部分署名）を付与することがある。

#### 【0009】

今後は公開鍵暗号基盤の普及によりこのような電子取引が主流になると考えられている。

#### 【0010】

一般にある文書構造定義言語で記述された文書構造定義を別の言語で書かれた等価な文書構造定義に変換する技術が存在する。例えば、非特許文献3には文書定義言語DTDで記述された文書構造定義を「W3C XML Schema」で記述された等価な文書構造定義に変換するツール「dtd2xs」が紹介されている。ただし、文書構造定義言語の記述能力の違いなどに起因して、あらゆる文書構造定義言語間の変換で完全に等価な文書構造定義が得られると言うわけではない。

#### 【0011】

##### 【非特許文献1】

Rajiv Mordani、他1名、「Java API for XML Processing Version 1.2 Final Release」、[online]、2003年2月10日、[同日検索]、インターネット<URL: <http://www.jcp.org/aboutJava/communityprocess/final/jsr063/index2.html>>

##### 【非特許文献2】

The Apache Software Foundation、「Xerces2 Java Parser Readme」、[online]、2002年、[2003年2月10日検索]、インターネット<URL: <http://xml.apache.org/xerces2-j/index.html>>

##### 【非特許文献3】



LuMriX、「XML Tools」、[online]、2003年、[2003年2月14日検索]、インターネット<URL:http://puvogel.informatik.med.uni-giessen.de/lumrix/#dtd>

#### 【0012】

##### 【発明が解決しようとする課題】

従来技術では、前記のように送受信される構造化文書に署名付与、暗号化などの処理を施すと文書の構造が変化するため、署名付与、暗号化などの処理を施さない文書と同じ方法で構造検査を実施することができない、という問題点がある。署名付与だけの場合、署名全体を除去してから構造検査を行うようにすることも可能だが、上記暗号化されたクレジットカード情報を受け取ったオンラインショッピング運営者のように、自身では復号化できない文書を受け取った場合、構造の検査が実施できなくなる。

#### 【0013】

本発明の目的は、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施可能とすることにある。

#### 【0014】

##### 【課題を解決するための手段】

本発明では、文書に対する署名付与の場所と方式と署名データの構造、暗号化対象のデータと暗号化方式など文書構造を変化させる処理の内容をすべて事前に取り決め、文書構造検査時に利用される文書構造定義を検査前に署名付与／暗号化後の形式に変換し、該変換後の文書構造定義を利用して従来どおりの文書構造検査を実施することにより前記目的を達成する。

また、本発明では別の方法として、文書構造検査部が署名付与および暗号化による文書構造の変化規則を読み込み、さらに文書構造検査時に該変化規則を動的に適用するように文書構造検査部を拡張して、送信者の署名の有無／方式、暗号化箇所／方式などを認識することにより、前記目的を達成する。

## 【0015】

## 【発明の実施の形態】

以下の説明では構造化文書の、署名および暗号化による構造の変化に対応した文書構造検査方法および装置について述べているが、本発明は署名および暗号化以外の操作による文書構造変化にも対応可能である。

## 【0016】

以下、本発明の第1の実施例を図により説明する。

本発明は図1に示したような複数の計算機101、102、104がネットワーク103（有線または無線）により接続され、相互に情報のやり取りを行う環境において、情報の受信者が、受信データ構造が事前に取り決めたものと一致しているかどうかを検査する方法およびシステムに関するものである。

## 【0017】

本実施例では2者間の情報のやり取りを例にとって説明するが、本発明は一連の処理に3者以上が関わる通信に含まれる任意の2者間通信にも適用可能である。すなわち、AからBを経てCに情報が送られるような3者が関わる通信の場合、AからBへの送信、および、BからCへの送信の2箇所でも適用可能である。

## 【0018】

図2は受信者側の計算機の構成を示したものである。受信者側の計算機104はCPU201、メモリ202、表示装置203、入力装置204、通信装置205、記憶装置206からなり、ネットワーク103に接続されている。該記憶装置206には受信文書の構造を検査する文書構造検査部211、受信文書に含まれる業務要求を処理する業務処理部212、受信文書の構造を記述した文書構造定義を記憶する文書構造定義集213、署名や暗号化など、送信文書に対する操作によりどのように文書構造が変化するかを記述した文書構造変化規則を記憶する文書構造変化規則集215、前記文書構造変化規則に従って前期文書構造定義を変化させる文書構造定義変換部214が含まれる。

## 【0019】

本発明は記憶装置に格納された前記処理プログラム（文書構造検査部、業務処理部、文書構造定義変換部）がメモリに読み込まれ、CPUで実行されることに

より実現される。

#### 【0020】

本実施例の前記文書構造検査部 211 は非特許文献 2 で既に実現されているものと同じものである。すなわち、受信した構造化文書が対応する前記文書構造定義と合致するかどうかを検査する。この検査で合致した場合、前記業務処理部 212 で該受信文書の要求に従って業務処理を実行する。文書構造検査に先立って前記文書構造変換部 214 が前記文書構造変化規則に従って前記文書構造定義の変換処理を行う。前記文書構造検査部は該変換後の文書構造定義を利用して、検査を実施する。

#### 【0021】

図 3 は前記文書構造定義集 213 に記録される文書構造定義の一例を、DTD (Document Type Definition) を利用して示した例である。DTD は XML 文書などの文書構造を定義するために設計された文書構造定義言語である。ここでは DTD を利用した場合の例を示したが、他の文書構造定義言語による定義でも構わない。XML 文書は入れ子関係を持つ要素の集合で、その先頭の要素を「ルート要素」と呼ぶ。要素 E が要素 F に含まれる場合、要素 E と要素 F の関係を「親子関係」と呼び、要素 E は要素 F の「親要素」、要素 F は要素 E の「子要素」と呼ぶ。同じ親要素を持つ要素同士を「兄弟要素」と呼ぶ。各要素はそれぞれ 0 個以上の属性を持つことができる。

#### 【0022】

図 3 には説明のために文書構造定義の各行の先頭に 2 桁の行番号を付与しているが、実際の文書構造定義には行番号は含まれない。同様に、以降の説明で利用する構造化文書、文書構造定義には説明のために行番号を付与しているが実際の構造化文書、文書構造定義にはこの番号は含まれない。

#### 【0023】

図 3 の 1 行目 (行番号 01) はルート要素が「PurchaseOrder」であることを表す。2 行目 (行番号 02) は前記「PurchaseOrder」要素が「UserID」、「Price」、「CreditCard」の 3 つの子要素を持ち、これらがこの順で 1 回だけ出現することを表す。3 行目 (行番

号03)は前記「PurchaseOrder」要素が「Id」という属性を持ち、該属性が存在する場合、その値がXML文書中で一意でないといけないこと、該属性は存在しなくても良いことを表す。4行目(行番号04)、5行目(行番号05)はそれぞれ「UserID」要素、「Price」要素の値として任意の文字列が許容されることを表す。6行目(行番号06)は「CreditCard」要素が「Issuer」、「Number」、「Expire」、「Owner」の4つの子要素を持ち、これらがこの順で1回だけ出現することを表す。7行目(行番号07)、8行目(行番号08)、9行目(行番号09)、10行目(行番号10)はそれぞれ「Issuer」要素、「Number」要素、「Expire」要素、「Owner」要素の値として、任意の文字列が許容されることを表す。ここでは、本文書構造定義を「PurchaseOrder.dtd」と名付けて前記文書構造定義集213に記憶しているものとする。

#### 【0024】

図4は前記文書構造定義「PurchaseOrder.dtd」に合致するXML文書の一例である。ただし、ここでは説明をわかりやすくするために、通常のXML文書に含まれる名前空間を省略している。名前空間はひとつのXML文書内で要素名を一意に保つために利用されるもので、本例では名前空間なしでも要素名が一意になるようにしているため名前空間を省略しても問題ない。

#### 【0025】

図4の1行目(行番号01)は本文書がXML文書であることを表し、2行目(行番号02)は本XML文書の構造が「PurchaseOrder.dtd」で定義されたものに合致することを表す。3行目(行番号03)以降が実際の文書を表す。4行目(行番号04)はUserID要素の値が「10194970」であることを、5行目(行番号05)はPrice要素の値が「100000」であることを表す。

#### 【0026】

図5は前記文書構造変化規則集215に記録される文書構造変化規則の例である。ここでは3つの規則511、512、513が挙げられており、それぞれの規則は、変換の方法を示す種別501、変換対象の文書構造定義と適用箇所を示

す適用場所 502、変化後の操作要素 503、変化後の文書構造定義の元となる関連文書構造定義 504 および文書構造変化規則番号 505 からなる。

#### 【0027】

前記種別 501 には「Replace」「Add」「Delete」の 3 種類がある。「Replace」は前記適用場所 502 で示した要素を前記操作要素 503 で示した要素で置換することを、「Add」は前記適用場所 502 で指定した場所に前記操作要素 503 を追加することを、「Delete」は前記適用場所 502 で示した要素が削除されることを表す。種別が「Delete」である場合、操作要素 503 および関連文書構造定義 504 は「空」としても構わない。

#### 【0028】

前記適用場所 502 は、文書構造定義とその定義内の場所を示す文字列を「:」記号で連結した文字列で表現される。該文書構造定義内の場所は木構造のルートから順に指定要素までの要素名を、「/」記号を区切り文字として連結した文字列で表現される。

#### 【0029】

また、ある要素を指定するのではなく木構造の特定の位置を指定するために、「ある要素の先頭の子要素のさらに前」を意味する「first ()」や、「ある要素の末尾の子要素のさらに後」を意味する「last ()」、「ある要素の直前（ある要素 E の直前に兄弟要素 D が存在する場合 E と D の間）」を意味する「before ()」、「ある要素の直後（ある要素 E の直後に兄弟要素 F が存在する場合 E と F の間）」を意味する「after ()」を利用して表現しても良い。「ある要素の値」を意味する「value ()」を利用して表現しても良い。さらに、連続する兄弟要素を記号「-」で連結して、範囲を表現することも可能である。例えば「po. dtd: / po / value ()」は文書構造定義「po. dtd」のルート要素「po」の値を、「po. dtd: / po / one - three」は文書構造定義「po. dtd」のルート要素「po」の連続する子要素「one」から「three」までの範囲を表す。

#### 【0030】

例えば、1番目の文書構造変化規則 511 の適用場所 502 「PurchaseOrder.dtd:/PurchaseOrder/CreditCard」は文書構造定義「PurchaseOrder.dtd」のルート要素「PurchaseOrder」の子要素「CreditCard」を表す。また、2番目の文書構造変化規則 512 の適用場所 502 「PurchaseOrder.dtd:/PurchaseOrder/last ()」は文書構造定義「PurchaseOrder.dtd」のルート要素「PurchaseOrder」の最後の子要素のさらに後を表す。ここでは特殊な表記法を用いたが、文書構造定義の特定の場所を指し示すことができる表記法であれば上記以外の方法で表現しても良い。

#### 【0031】

前記操作要素 503 も前記適用場所と同じ表記方法で置換または追加する要素を表現する。例えば、1番目の文書構造変化規則 511 の操作要素「EncryptedData.dtd:/EncryptedData」は文書構造定義「EncryptedData.dtd」のルート要素「EncryptedData」を表す。

#### 【0032】

前記関連文書構造定義 504 は文書構造の変化時に必要となる文書構造定義を表す。複数の文書構造定義をあげることができる。例えば、1番目の文書構造変化規則 511 の関連文書構造定義「EncryptedData.dtd KeyInfo.dtd」は「EncryptedData.dtd」と「KeyInfo.dtd」の2つの文書構造定義が該文書構造変化規則に関連していることを示す。該関連文書構造定義 504 であげた文書構造定義を含めて、前記文書構造変化規則集 215 内に出現する全ての文書構造定義を事前に入手し、前記文書構造定義集 213 にあらかじめ記憶していなければならない。

#### 【0033】

図6はEncryptedData.dtdの例、図7はEncryptedKey.dtdの例、図8はSignature.dtdの例、図9はKeyInfo.dtdの例である。それぞれはW3C (World Wide Web

Consortium)で策定されたXML文書の暗号化に関する仕様(XMLEncryption)および、XML文書の署名に関する仕様(XMLSignature)を元に、説明のために簡単化して記述したものである。

#### 【0034】

図10は前記文書構造定義変換部214の処理手順を表す。はじめに全ての文書構造変化規則を適用済みかどうかを検査し(ステップ1001)、全て適用済みなら処理を終了する。未適用の変化規則がある場合、次の変化規則を取得し(ステップ1002)、種別を調べる。種別が「Replace」の場合(ステップ1003)、該変化規則の適用場所に記載の要素を操作要素で置換する(ステップ1010)。種別が「Add」の場合(ステップ1004)、該変化規則の適用場所に操作要素を追加する(ステップ1011)。

さらに、前記置換処理(ステップ1010)および追加処理(ステップ1011)にはこれに伴って必要となる、前記文書構造定義集213への、置換および追加要素とその定義に利用される要素の構造定義の追加も行う(ステップ1012)。

#### 【0035】

図5の3つの文書構造変化規則511、512、513を適用する例を以下に示す。

図3の文書構造定義「PurchaseOrder.dtd」に、1番目の文書構造変化規則511を適用した後の状態を図11に示す。1番目の文書構造変化規則511は種別が「Replace」、適用場所が「PurchaseOrder.dtd:/PurchaseOrder/CreditCard」、操作要素が「EncryptedData.dtd:/EncryptedData」なので、「PurchaseOrder.dtd」のルート要素「PurchaseOrder」の子要素「CreditCard」(図3の320)を「EncryptedData.dtd」のルート要素「EncryptedData」(1121)で置換する。変換前、ルート要素「PurchaseOrder」は「UserID」「Price」「CreditCard」の3つの子要素を持つと定義されている(図3の行番号02)。これが変換により「User I



D) 「Price」 「EncryptedData」 の3つの子要素を持つようになる (図11の行番号02)。

#### 【0036】

さらに、1番目の文書構造変化規則511の関連文書構造定義にあげられた2つの文書構造定義「EncryptedData.dtd」および「KeyInfo.dtd」を適用対象となる文書構造定義「PurchaseOrder.dtd」に追加するのが図11の行番号11~16 (1114)、及び図11の行番号17~22 (1115) である。この例ではこれらの文書構造定義の実体を追加しているが、対応する文書構造定義の参照 (文書構造定義を一意に特定するための情報) のみを追加し、実体を前記文書構造定義集213に「PurchaseOrder.dtd」とは分離して記憶する方式でも良い。図22に示した文書構造定義2201は、図11に示した文書構造定義1101を文書構造定義「EncryptedData.dtd」と「KeyInfo.dtd」への参照を用いて表現した例である。図11の行番号11~16 (1114) に対応する部分を参照を用いて表現したのが図22の行番号11~12 (2211)、図11の行番号17~22 (1115) に対応する部分を参照を用いて表現したのが図22の行番号13~14 (2212) である。

#### 【0037】

また、要素の置換処理により「CreditCard」要素は不要となるためその定義 (行番号06) および、「CreditCard」の定義でのみ利用される他の要素 (図11の行番号07~10 (1113)) は文書構造定義「PurchaseOrder.dtd」から削除しても良い。

#### 【0038】

次に、さらに2番目の文書構造変化規則512を適用した後の文書構造定義「PurchaseOrder.dtd」を図12に示す。2番目の文書構造変化規則512は種別が「Add」、適用場所が「PurchaseOrder.dtd: /PurchaseOrder/last ()」、操作要素が「EncryptedKey.dtd: /EncryptedKey」なので、「PurchaseOrder.dtd」のルート要素「PurchaseOrder」の最



後の要素として「EncryptedKey. dtd」のルート要素「EncryptedKey」を追加する（図12の行番号02の1211）。さらに、2番目の文書構造変化規則の関連文書構造定義にあげられた文書構造定義「EncryptedKey. dtd」を適用対象となる文書構造定義「PurchaseOrder. dtd」に追加する（図12の行番号23～29（1212））。

#### 【0039】

最後に、同様にしてさらに3番目の文書構造変化規則513を適用した後の文書構造定義「PurchaseOrder. dtd」を図13に示す。「PurchaseOrder」要素の最後の要素として「Signature. dtd」のルート要素「Signature」を追加し（行番号02の1311）、「Signature」要素の構造定義（行番号30～41の1312）を追加している。

#### 【0040】

以上で文書構造変化規則集215の全ての文書構造変化規則を適用したため、前記文書構造変換部214の処理を終了する。

#### 【0041】

上記の文書構造変換処理により、図4に示した構造化文書を暗号化し、署名を付与した後の文書（図14）の構造検査が可能となる。図14の行番号06～15（1411）は「CreditCard」要素を暗号化した結果、行番号16～27（1412）は暗号に利用した鍵の情報、行番号28～41（1413）は署名を表す。

#### 【0042】

上記では文書構造変化規則として、「PurchaseOrder. dtd」の構造が変化するもののみをあげたが、図15に示すような「PurchaseOrder. dtd」以外の文書構造定義の構造を変化させる文書構造変化規則を定義することも可能である。

#### 【0043】

図15の1511は署名の方式の一つで、署名対象文書を署名の中に取り込む

方式（一般に「Enveloping 署名」と呼ばれている）に対応させる文書構造変化規則の例である。

#### 【0044】

これを適用した場合の文書構造定義の変化を図16に示す。文書構造変化規則1511は種別が「Add」、適用場所が「Signature.dtd:/Signature/last()」、操作要素が「PurchaseOrder.dtd:/PurchaseOrder」なので、文書構造定義「Signature.dtd」のルート要素「Signature」の最後の子要素として、文書構造定義「PurchaseOrder.dtd」のルート要素「PurchaseOrder」を行番号02に追加(1611)、さらに、関連文書構造定義にあげられた「PurchaseOrder.dtd」と「KeyInfo.dtd」を「Signature.dtd」に行番号14~22(1612)、及び行番号23~28(1613)で追加する。該文書構造変化規則を適用すると受信文書のルート要素は「Signature」に変更されることになる。

#### 【0045】

上記のように、本発明の第1の実施例によれば、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施することが可能となる。

#### 【0046】

以下、本発明の第2の実施例を図により説明する。

図17に本発明の第2の実施例のハードウェア構成を示す。本発明の第2の実施例のハードウェア構成は、第1の実施例と同じである。違いは記憶装置内の文書構造検査部1701での処理方法、および、文書構造変化規則集1711のデータ構造にある。

#### 【0047】

図18は前記文書構造変化規則集1711に記録される文書構造変化規則の例である。ここでは4つの規則1811、1812、1813、1814があげら

れており、それぞれの規則は、変換の方法を示す種別 1801、適用文書構造定義を表す適用先 1802、変換操作の対象となる操作要素 1803、変化後の文書構造検査に必要となる関連文書構造定義 1804 および文書構造変化規則番号 1805 からなる。

#### 【0048】

前記種別 1801 には「Replace」「Add」の 2 種類がある。「Replace」は前記適用先 1802 の文書構造定義の一部が前記操作要素 1803 で示した文書構造定義の要素に置換される可能性があることを、また「Add」は前記適用先 1802 の文書構造定義の一部に前記操作要素 1803 で示した文書構造定義の要素が追加される可能性があることを表す。前記適用先 1802 には文書構造定義名または記号「\*」が記述される。「\*」は適用先としてどの文書構造定義でも良いことを表す。前記操作要素 1803 および関連文書構造定義 1804 の表記法はそれぞれ第 1 の実施例で示した図 5 の操作要素 503 および関連文書構造定義 504 に準ずる。

#### 【0049】

図 19 は前記文書構造検査部 1701 の処理手順を表す。はじめに検査対象となる構造化文書に記載された対応文書構造定義を特定し、現文書構造定義に設定する（ステップ 1901）。検査対象構造化文書の検査位置および文書構造定義の検査位置は共に先頭に設定する。次に、現文書構造定義と検査対象構造化文書が合致するかどうかを検査位置から順番に従来通り検査する（ステップ 1902）。ステップ 1902 は検査が文書構造定義の末尾に到達する、もしくは、検査対象構造化文書の末尾に到達する、もしくはエラーが発生する、のいずれかの場合に停止してステップ 1903 に処理が移る。ここで「文書構造定義の末尾」とは文書構造定義の最終行を意味するわけではなく、「当該文書構造定義のルート要素の定義の終端」を意味する。

#### 【0050】

ステップ 1903 では検査が文書構造定義の末尾まで到達したかどうかを調べ、検査が末尾に到達しなかった場合、異常終了の原因を見極める。すなわち、異常の原因が「不正要素の出現」であったかどうか調べ（ステップ 1904）、そ

うでなかった場合に「不一致」との結果を返す。前記異常が「不正要素の出現」であった場合、文書構造変化規則集から適用可能な文書構造変化規則を検索する（ステップ1905）。ここで「適用可能」とは、現文書構造定義が文書構造変化規則の「適用先」に合致し、かつ、検査対象構造化文書に現れた不正要素が文書構造変化規則の「操作要素」に一致することを意味する。ただし、「適用先」が「\*」の場合、あらゆる文書構造定義に合致するものとする。

#### 【0051】

ステップ1905で適用可能な文書構造定義が発見（ステップ1906）できなかった場合、「不一致」との結果を返す。ステップ1905で適用可能な文書構造変化規則を発見した場合、現文書構造定義（ここでは名称）とその不一致箇所（ここでは第1の実施例の文書構造変化規則の適用場所502と同じ表記方法を用いて表現している）、発見した文書構造変化規則の種別の3つの情報をひとまとまりにしてスタックにP u s hし、発見した文書構造変化規則の操作要素の文書構造定義を現文書構造定義に設定して（ステップ1907）、検査を続行する（ステップ1902へ）。

ステップ1903で正常に終了した場合、スタックの状態を調べ（ステップ1908）、空の場合、検査対象構造化文書の末尾まで到達（ステップ1912）していれば「一致」を返して正常終了する。ステップ1912で検査対象構造化文書の末尾に到達していなければ「不一致」を返して終了する。ステップ1908でスタックが空でない場合、スタックの先頭から文書構造定義、不一致箇所、変化規則の種別をP o pし、P o pした文書構造定義を現文書構造定義に設定する（ステップ1909）。次に、ステップ1909でP o pした変化規則の種別が「R e p l a c e」であるかどうか調べ（ステップ1910）、「R e p l a c e」でない場合、検査を続行する（ステップ1902へ）。ステップ1909でP o pした変化規則が「R e p l a c e」であった場合、現文書構造定義の検査位置を次の兄弟要素にする。次の兄弟要素が存在しない場合は、現要素の直後に検査位置を移動する（ステップ1911）。ステップ1909でP o pした変化規則が「R e p l a c e」でなかった場合、および、ステップ1911の実行後はステップ1902へ進み、検査を続行する。



## 【0052】

図14の構造化文書を検査対象として、図19の手順に従って文書構造検査を行う様子を以下で示す。

## 【0053】

文書構造定義集213には図3の「PurchaseOrder.dtd」、図6の「EncryptedData.dtd」、図7の「EncryptedKey.dtd」、図8の「Signature.dtd」、図9の「KeyInfo.dtd」が、文書構造変化規則集1711には図18の4つの規則が記憶されているものとする。

## 【0054】

はじめに、ステップ1901で検査対象構造化文書（図14の1401）の2行目から文書構造定義を「PurchaseOrder.dtd」に特定する。次にステップ1902で、エラー発生もしくは検査終了まで通常通りの検査を行う。この例では3行目から5行目までは正常に検査が行われるが、6行目の要素「EncryptedData」まで来たところで、文書構造定義（図3）の2行目に記載されている要素「CreditCard」と一致しないためエラーが発生する。このエラーは不正要素「EncryptedData」の出現であるためステップ1905へ進み、文書構造変化規則集から適用先が現文書構造定義「PurchaseOrder.dtd」と合致し、かつ、操作要素が「EncryptedData」と一致する文書構造変化規則を検索する。検索の結果、図18に示す1番目の文書構造変化規則1811が見つかる。1番目の文書構造変化規則1811の適用先は「\*」であるが、これはどの文書構造変化規則とも合致するので「PurchaseOrder.dtd」にも合致する。

## 【0055】

次にステップ1907で現文書構造定義「PurchaseOrder.dtd」、不一致箇所「/PurchaseOrder/CreditCard」、発見した文書構造変化規則1の種別「Replace」をスタックにPushする。その様子を図20に示す。スタックの先頭（最上部）に「PurchaseOrder.dtd|/PurchaseOrder/CreditCard|

Replace」(2001)が記憶される。次に、ステップ1905で発見した文書構造変化規則1の操作要素「EncryptedData.dtd:/EncryptedData」を現文書構造定義に設定し、さらに検査を続ける(ステップ1902へ)。

#### 【0056】

次に、ステップ1902で図6の文書構造定義「EncryptedData.dtd」601の「EncryptedData」要素(行番号02)と、図14の検査対象構造化文書1401の行番号06以降の検査が行われる。本検査は図14の検査対象構造化文書1401の行番号15の終わりまで進むと、文書構造定義601の末尾まで到達するのでステップ1903で「はい」の方に進み、スタックが検査される。スタックには先ほどステップ1907でPushした情報2001が記憶されており空ではないので、ステップ1909へ進み、スタックの先頭の情報2001をPopする。この処理によりスタックは空になる。さらに、現文書構造定義をさきほどPopした文書構造定義「PurchaseOrder.dtd」に、検査位置を「/PurchaseOrder/CreditCard」に設定する。次に、ステップ1909でPopした文書構造変化規則の種別が「Replace」のため、ステップ1911へ進む。ステップ1911ではステップ1909で「/PurchaseOrder/CreditCard」に設定した現文書構造定義の検査位置を次の兄弟要素に設定する。ここではCreditCard要素に次の兄弟要素が存在しないので、「CreditCard要素の直後(/PurchaseOrder/CreditCard/after())」へ検査位置を設定する。その後、ステップ1902へ進み、さらに検査を続ける。

#### 【0057】

ここまでで現文書構造定義「PurchaseOrder.dtd」の「/PurchaseOrder/CreditCard/after()」に検査位置が進み、検査対象構造化文書(図14の1401)の検査位置は行番号15の終わりまで進んでいる。

#### 【0058】



ステップ1902で検査を続行すると、検査対象構造化文書（図14の1401）の行番号16に不正要素「EncryptedKey」が出現することがわかるので、ステップ1905へ進み、適用先と操作要素一致する文書構造変化規則を検索する。今回の検索では、図18に示した2番目の文書構造変化規則1812が見つかるのでスタックに「PurchaseOrder.dtd|PurchaseOrder/CreditCard/after()|Add」をPushする。図21にこの時のスタックの状態を示す。

さらに、現文書構造定義を「EncryptedKey.dtd」、検査位置を「EncryptedKey要素」に設定して、ステップ1902へ進み、検査を続ける。

今回のステップ1902では図14の検査対象構造化文書1401の行番号27の終わりまで検査が進んだところで、現文書構造定義「EncryptedKey.dtd」の末尾に到達するため、処理がステップ1908へ進む。次に、スタックの先頭の情報「PurchaseOrder.dtd|2行目CreditCard要素の直後|Add」をPopし、現文書構造定義および検査位置をそれぞれ「PurchaseOrder.dtd」「PurchaseOrder/CreditCard/after()」に設定する。今回Popした種別は「Add」なので、ステップ1902へ進み、検査を続ける。

#### 【0059】

ステップ1902で検査を続行すると、検査対象構造化文書の検査位置は変化しないまま、不正要素「Signature」が出現するため、ステップ1905へ進み、適用先と操作要素一致する文書構造変化規則を検索する。今回の検索では、図18に示した3番目の文書構造変化規則1813が見つかるのでスタックに「PurchaseOrder.dtd|PurchaseOrder/CreditCard/after()|Add」をPushする。さらに、現文書構造定義を「Signature.dtd」、検査位置を「Signature要素」に設定して、ステップ1902へ進み、検査を続ける。

#### 【0060】

今回のステップ1902では検査対象構造化文書1401の行番号41の終わ

りまで検査が進んだところで、現文書構造定義「Signature. dtd」の末尾に到達するため、処理がステップ1908へ進む。次に、スタックの先頭の情報「PurchaseOrder. dtd | /PurchaseOrder /CreditCard /after () | Add」をPopし、現文書構造定義および検査位置をそれぞれ「PurchaseOrder. dtd」「/PurchaseOrder /CreditCard /after ()」に設定する。この処理によりスタックは空になる。今回Popした種別は「Add」なので、ステップ1902へ進み、検査を続ける。

#### 【0061】

ステップ1902で検査が、検査対象構造化文書1401の行番号42の終わりまで到達すると、現文書構造定義「PurchaseOrder. dtd」の末尾に到達するのでステップ1908へ進む。今回はスタックは空で、かつ、検査対象構造化文書の末尾（行番号42の終わり）まで到達しているので、検査を終了して「一致」を返す。

#### 【0062】

以上で検査対象構造化文書1401の検査が終了する。

検査対象構造化文書1401の検査では、4番目の文書構造変化規則1814を適用しなかった。4番目の文書構造変化規則は署名対象文書が署名自体に内包される構造化文書（図16の文書構造定義に合致する）の検査の際に利用するものである。

上記のように、本発明の第2の実施例によれば、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施することが可能となる。

#### 【0063】

第1および第2の実施例では全ての文書構造定義を1種類の文書構造定義言語DTDで記述した場合を想定していたが、第1の実施例では「文書構造定義への参照（2211、2212）を利用してその実体を取得する時」に、また、第2の実施例では「ステップ1902を実行する際」に、文書構造定義の記述に利用



されている文書構造定義言語の種別を判断し、対応する文書構造検査部を選択するようにすることで、複数の文書構造定義言語を利用して文書構造定義を記述した場合でも検査対象構造化文書の構造検査を実施することができるようになる。前記「文書構造定義言語の種別の判断」は文書構造定義名で行うことができる。

#### 【0064】

例えば、文書構造定義名「PurchaseOrder.dtd」はその拡張子「.dtd」で、該文書構造定義がDTDにより記述されていることを、また、文書構造定義名「Signature.xsd」はその拡張子「.xsd」で文書構造定義が「W3C XML Schema」により記述されていることを意味することができる。

#### 【0065】

さらに、複数の文書構造定義言語を利用して文書構造定義を記述した場合でも検査対象構造化文書の構造検査を実施可能とする別の方法として、文書構造定義集213に記憶された文書構造定義を1種類の文書構造定義言語を用いた表現に変換する方法も適用可能である。

#### 【0066】

##### 【発明の効果】

本発明により、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、暗号化部分の検査者の復号化可否に関わらず、文書全体の構造検査を実施することが可能となる。

##### 【図面の簡単な説明】

##### 【図1】

計算機環境を表す図である。

##### 【図2】

実施例1の計算機内部のハードウェア構成を表す図である。

##### 【図3】

文書構造定義の例である。

##### 【図4】

構造化文書の例である。

【図 5】

実施例 1 の文書構造変化規則集の例である。

【図 6】

文書構造定義の例である。

【図 7】

文書構造定義の例である。

【図 8】

文書構造定義の例である。

【図 9】

文書構造定義の例である。

【図 1 0】

実施例 1 の処理手順を表す図である。

【図 1 1】

文書構造変化規則 1 を適用した後の文書構造定義である。

【図 1 2】

文書構造変化規則 1 と 2 を適用した後の文書構造定義である。

【図 1 3】

文書構造変化規則 1 と 2 と 3 を適用した後の文書構造定義である。

【図 1 4】

構造変化後の構造化文書の例である。

【図 1 5】

文書構造変化規則の例である。

【図 1 6】

文書構造変化規則 4 を適用した後の文書構造定義である。

【図 1 7】

実施例 2 の計算機内部のハードウェア構成を表す図である。

【図 1 8】

実施例 2 の文書構造変化規則集の例である。

**【図 1 9】**

文書構造検査の詳細な処理手順を表す図である。

**【図 2 0】**

スタックの状態を表す図である。

**【図 2 1】**

スタックの状態を表す図である。

**【図 2 2】**

参照を利用して記述した文書構造定義の例である。

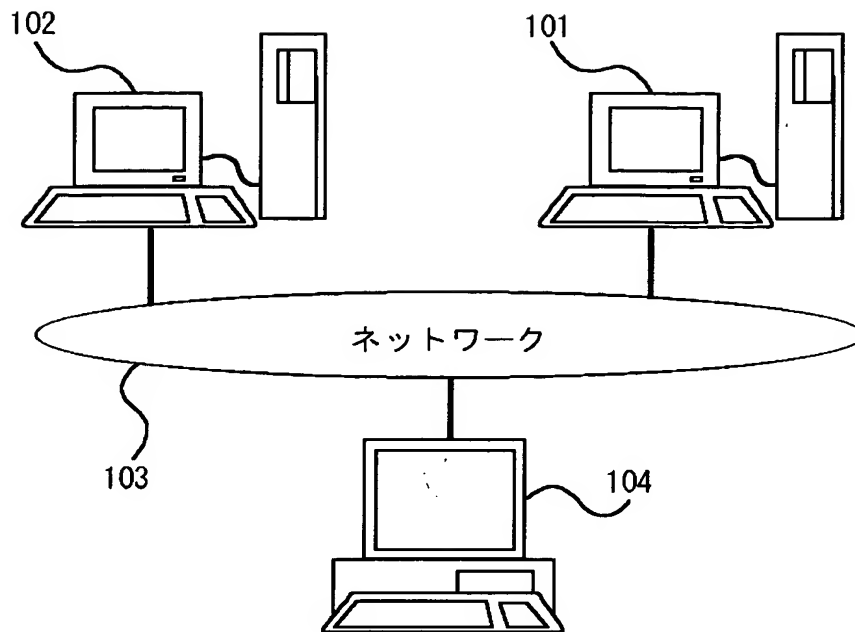
**【符号の説明】**

1 0 1 . . . . . 計算機、  
1 0 2 . . . . . 計算機、  
1 0 3 . . . . . ネットワーク、  
1 0 4 . . . . . 受信者側の計算機、  
2 0 1 . . . . . C P U、  
2 0 2 . . . . . メモリ、  
2 0 3 . . . . . 表示装置、  
2 0 4 . . . . . 入力装置、  
2 0 5 . . . . . 通信装置、  
2 0 6 . . . . . 記憶装置、  
2 1 1 . . . . . 文書構造検査部、  
2 1 2 . . . . . 業務処理部、  
2 1 3 . . . . . 文書構造定義集、  
2 1 4 . . . . . 文書構造定義変換部、  
2 1 5 . . . . . 文書構造変化規則集、  
1 7 0 1 . . . . . 文書構造検査部、  
1 7 1 1 . . . . . 文書構造変化規則集。

【書類名】 図面

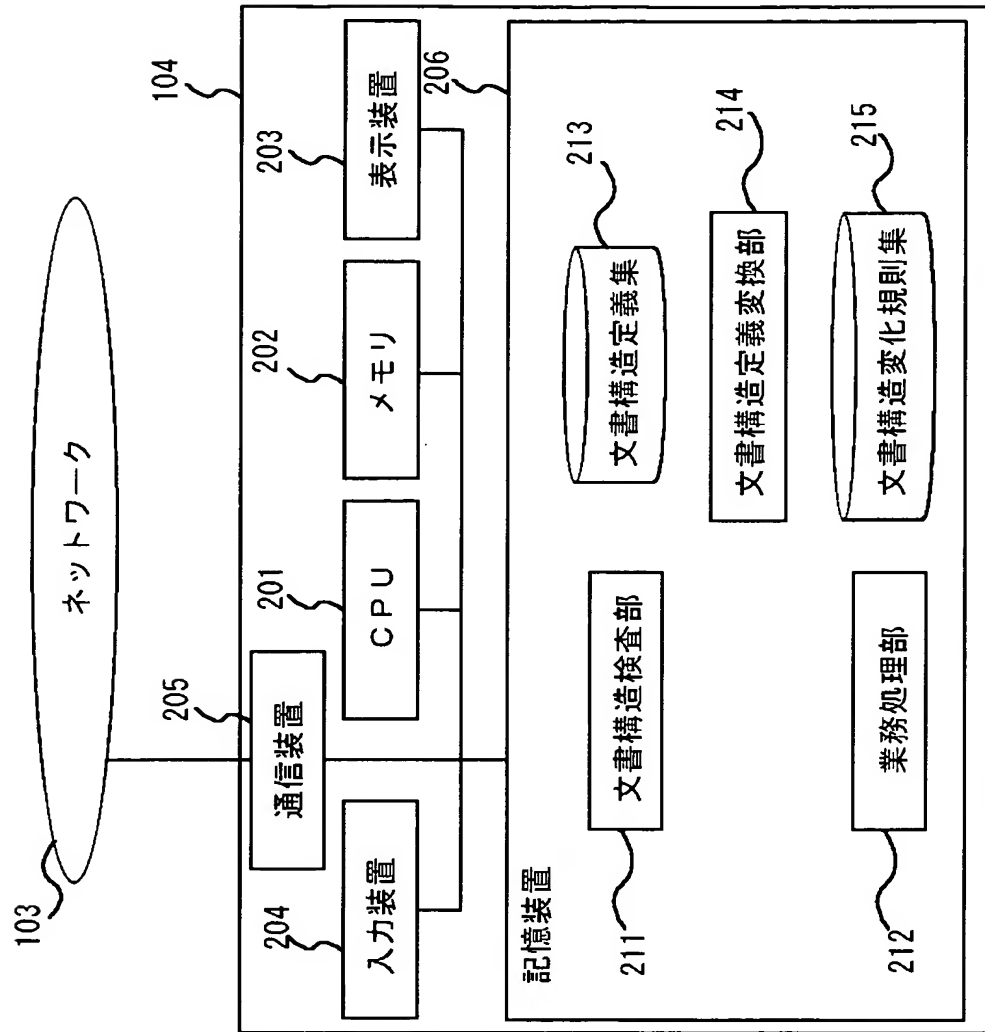
【図 1】

図 1



【図 2】

図 2



【図3】

図 3

```
01 <!DOCTYPE PurchaseOrder[
02   <!ELEMENT PurchaseOrder (UserID, Price, CreditCard)>
03     <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04   <!ELEMENT UserID (#PCDATA)>
05   <!ELEMENT Price (#PCDATA)>
06   <!ELEMENT CreditCard (Issure, Number, Expire, Owner)>
07     <!ELEMENT Issure (#PCDATA)>
08     <!ELEMENT Number (#PCDATA)>
09     <!ELEMENT Expire (#PCDATA)>
10     <!ELEMENT Owner (#PCDATA)>
11]>
```

320

【図 4】

図 4

```
01 <?xml version="1.0"?>
02 <!DOCTYPE PurchaseOrder SYSTEM "PurchaseOrder.dtd">
03 <PurchaseOrder>
04   <UserID>10194970</UserID>
05   <Price>100000</Price>
06   <CreditCard>
07     <Issur>SDL</Issuer>
08     <Number>1234-5678-9012-3456</Number>
09     <Expire>12/05</Expire>
10     <Owner>Larry Gates</Owner>
11   </CreditCard>
12 </PurchaseOrder>
```

【図 5】

図 5

#	種別	適用場所	操作要素	関連文書構造定義
1	Replace	PurchaseOrder.dtd: /PurchaseOrder/CreditCard	EncryptedData.dtd: /EncryptedData	EncryptedData.dtd KeyInfo.dtd
2	Add	PurchaseOrder.dtd: /PurchaseOrder/last()	EncryptedKey.dtd: /EncryptedKey	EncryptedKey.dtd
3	Add	PurchaseOrder.dtd: /PurchaseOrder/last()	Signature.dtd: /Signature	Signature.dtd



図 6

【図 6】

601

```
01 <!DOCTYPE EncryptedData[
02 <!ELEMENT EncryptedData (EncryptionMethod, KeyInfo, CipherData)>
03 <!ATTLIST EncryptedData Id ID #REQUIRED>
04 <!ELEMENT EncryptionMethod (#PCDATA)>
05 <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
06 <!ELEMENT CipherData (CipherValue)>
07 <!ELEMENT CipherValue (#PCDATA)>
08 ]>
```

【図 7】

図 7

```
01<!DOCTYPE EncryptedKey [  
02 <!ELEMENT EncryptedKey (EncryptionMethod, KeyInfo, CipherData, ReferenceList)>  
03 <!--ATTLIST EncryptedKey Id ID #REQUIRED-->  
04 <!ELEMENT ReferenceList (DataReference | KeyReference)+>  
05 <!--ELEMENT DataReference(#PCDATA)>  
06 <!--ATTLIST DataReference URI CDATA #REQUIRED-->  
07 <!--ELEMENT KeyReference(#PCDATA)>  
08 <!--ATTLIST KeyReference URI CDATA #REQUIRED-->  
09 ]>
```

【図 8】

図 8

```
01<!DOCTYPE Signature [  
02  <!ELEMENT  Signature(SignedInfo, SignatureValue, KeyInfo?) >  
03  <!ELEMENT  SignedInfo(CanonicalizationMethod, SignatureMethod, Reference+) >  
04  <!ELEMENT  CanonicalizationMethod(#PCDATA) >  
05  <!ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED >  
06  <!ELEMENT  SignatureMethod(#PCDATA) >  
07  <!ATTLIST SignatureMethod Algorithm CDATA #REQUIRED >  
08  <!ELEMENT  Reference(DigestMethod, DigestValue) >  
09  <!ATTLIST Reference URI CDATA #REQUIRED >  
10  <!ELEMENT  DigestMethod(#PCDATA) >  
11  <!ATTLIST DigestMethod Algorithm CDATA #REQUIRED >  
12  <!ELEMENT  DigestValue(#PCDATA) >  
13  <!ELEMENT  SignatureValue(#PCDATA) >  
14 ] >
```

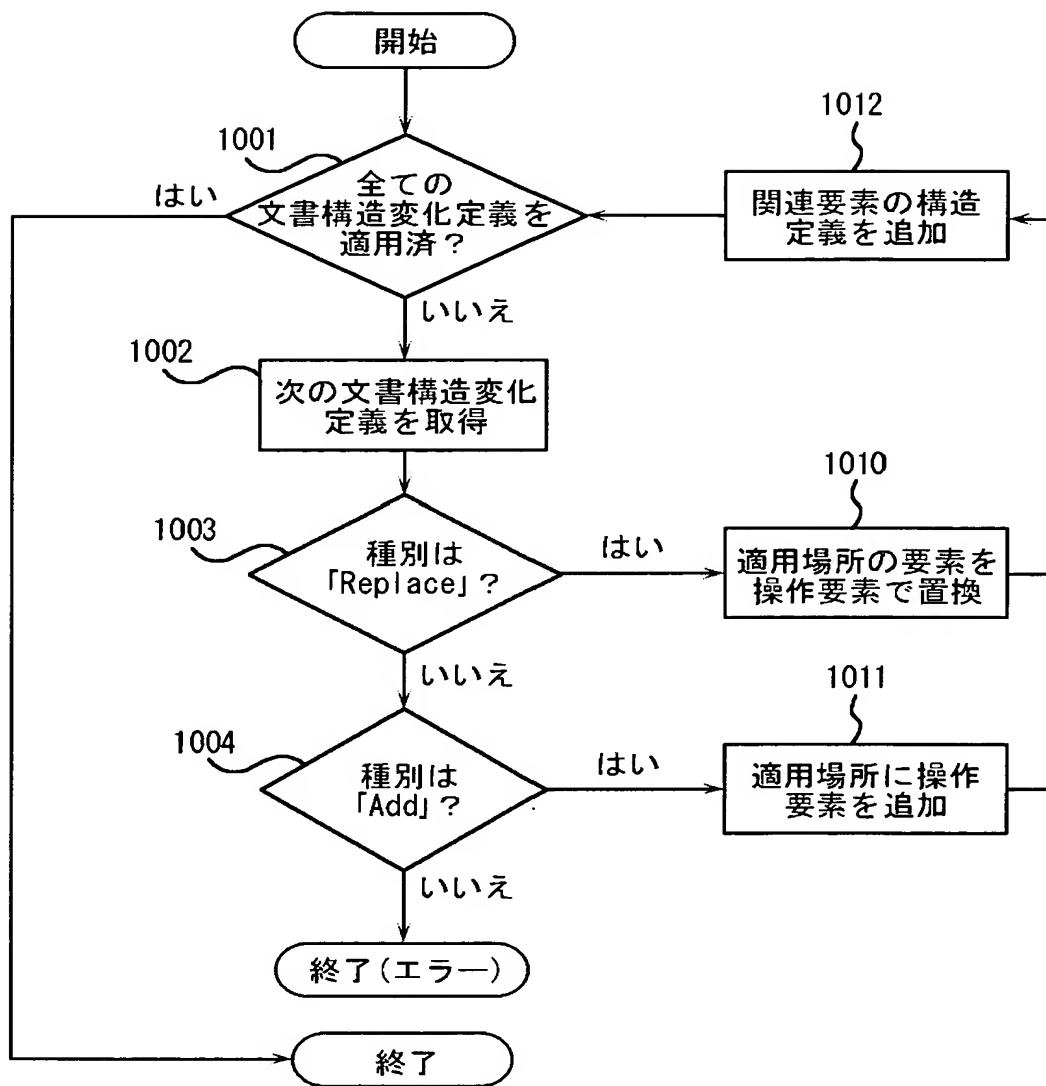
【図 9】

図 9

```
01 <!DOCTYPE KeyInfo[
02  <!ELEMENT KeyInfo(RetrievalMethod | KeyName)>
03  <!ELEMENT RetrievalMethod(#PCDATA)>
04    <!ATTLIST RetrievalMethod
05      Type          CDATA          #REQUIRED
06      URI           CDATA          #REQUIRED>
07  <!ELEMENT KeyName(#PCDATA)>
08 ]>
```

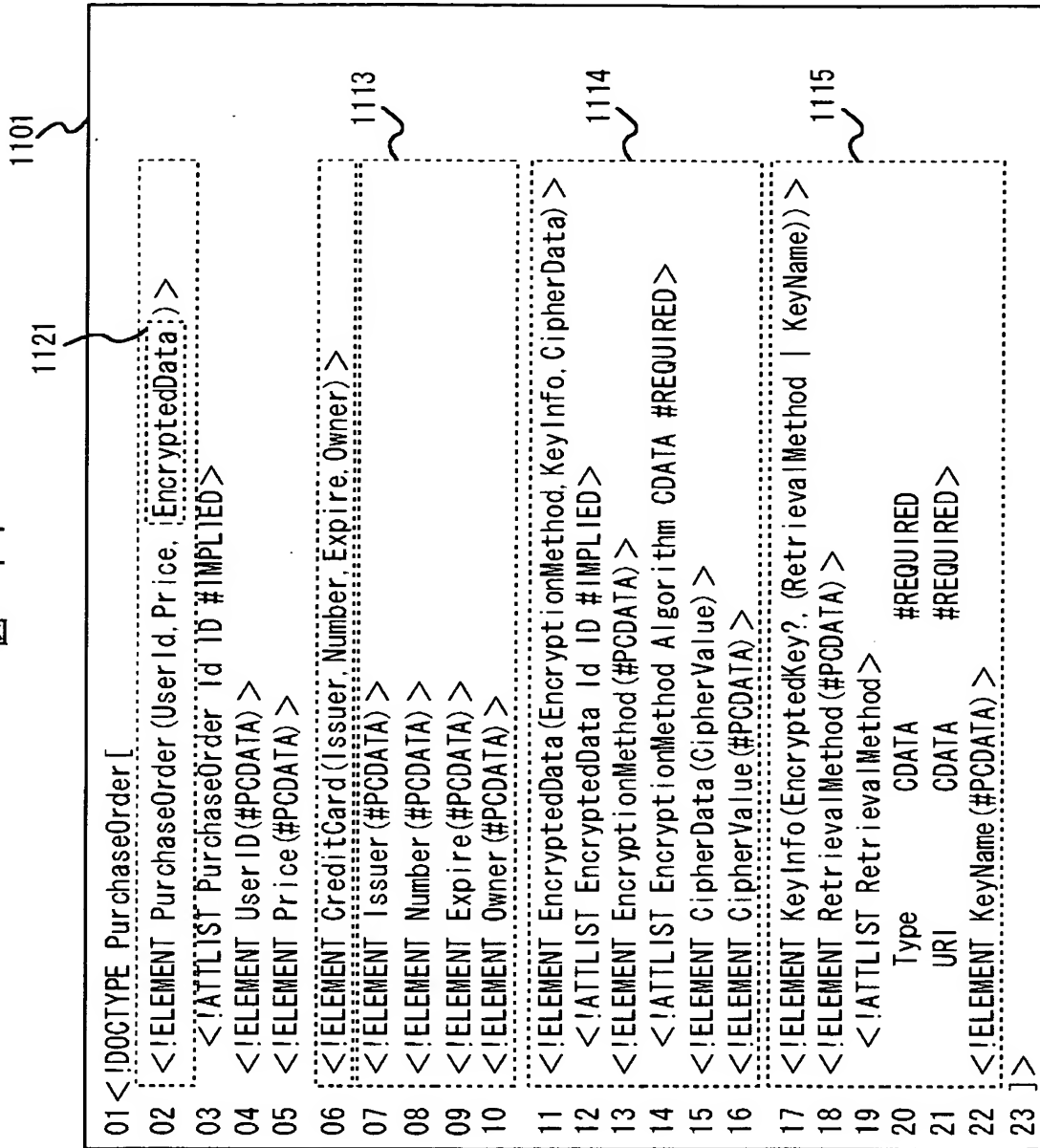
【図 10】

図 10



【図 11】

図 11



【図 12】

図 12

1201

```

01 <!DOCTYPE PurchaseOrder[
02   <!ELEMENT PurchaseOrder (UserID, Price, EncryptedData, EncryptedKey) >
03     <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04   <!ELEMENT UserID (#PCDATA) >
05   <!ELEMENT Price (#PCDATA) >
06   <!ELEMENT CreditCard (Issure, Number, Expire, Owner) >
07   <!ELEMENT Issure (#PCDATA) >
08   <!ELEMENT Number (#PCDATA) >
09   <!ELEMENT Expire (#PCDATA) >
10   <!ELEMENT Owner (#PCDATA) >
11   <!ELEMENT EncryptedData (EncryptionMethod, KeyInfo, CipherData) >
12     <!ATTLIST EncryptedData Id ID #IMPLIED>
13   <!ELEMENT EncryptionMethod (#PCDATA) >
14     <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
15   <!ELEMENT CipherData (CipherValue) >
16   <!ELEMENT CipherValue (#PCDATA) >
17   <!ELEMENT KeyInfo (EncryptedKey?, (RetrievalMethod | KeyName)) >
18   <!ELEMENT RetrievalMethod (#PCDATA) >
19     <!ATTLIST RetrievalMethod>
20       Type          CDATA          #REQUIRED
21       URI           CDATA          #REQUIRED
22   <!ELEMENT KeyName (#PCDATA) >
23   <!ELEMENT EncryptedKey (EncryptionMethod, KeyInfo, CipherData, ReferenceList) >
24     <!ATTLIST EncryptedKey Id ID #IMPLIED>
25   <!ELEMENT ReferenceList (DataReference | KeyReference)+ >
26   <!ELEMENT DataReference (#PCDATA) >
27     <!ATTLIST DataReference URI CDATA #REQUIRED>
28   <!ELEMENT KeyReference (#PCDATA) >
29     <!ATTLIST KeyReference URI CDATA #REQUIRED>
30 ]>

```

1211

1212

【図 13】

図 13

```

01 <!DOCTYPE PurchaseOrder[
02   <!--ELEMENT PurchaseOrder (User ID, Price, EncryptedData, EncryptedKey, Signature)-->
03   <!--ATTLIST PurchaseOrder Id ID #IMPLIED-->
04   <!--ELEMENT UserId(#PCDATA)-->
05   <!--ELEMENT Price(#PCDATA)-->
06   <!--ELEMENT CreditCard(Issure, Number, Expire, Owner)-->
07   <!--ELEMENT Issure(#PCDATA)-->
08   <!--ELEMENT Number(#PCDATA)-->
09   <!--ELEMENT Expire(#PCDATA)-->
10   <!--ELEMENT Owner(#PCDATA)-->
11   <!--ELEMENT EncryptedData(EncryptionMethod, KeyInfo, CipherData)-->
12   <!--ATTLIST EncryptedData Id ID #IMPLIED-->
13   <!--ELEMENT EncryptionMethod(#PCDATA)-->
14   <!--ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED-->
15   <!--ELEMENT CipherData(CipherValue)-->
16   <!--ELEMENT CipherValue(#PCDATA)-->
17   <!--ELEMENT KeyInfo(EncryptedKey?, (RetrievalMethod | KeyName))-->
18   <!--ELEMENT RetrievalMethod(#PCDATA)-->
19   <!--ATTLIST RetrievalMethod
20       Type          CDATA          #REQUIRED
21       URI            CDATA          #REQUIRED-->
22   <!--ELEMENT KeyName(#PCDATA)-->
23   <!--ELEMENT EncryptedKey(EncryptionMethod, KeyInfo, CipherData, ReferenceList)-->
24   <!--ATTLIST EncryptedKey Id ID #IMPLIED-->
25   <!--ELEMENT ReferenceList(DataReference | KeyReference)+-->
26   <!--ELEMENT DataReference(#PCDATA)-->
27   <!--ATTLIST DataReference URI CDATA #REQUIRED-->
28   <!--ELEMENT KeyReference(#PCDATA)-->
29   <!--ATTLIST KeyReference URI CDATA #REQUIRED-->
30   <!--ELEMENT Signature(SignedInfo, SignatureValue, KeyInfo?)-->
31   <!--ELEMENT SignedInfo(CanonicalizationMethod, SignatureMethod, Reference+)-->
32   <!--ELEMENT CanonicalizationMethod(#PCDATA)-->
33   <!--ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED-->
34   <!--ELEMENT SignatureMethod(#PCDATA)-->
35   <!--ATTLIST SignatureMethod Algorithm CDATA #REQUIRED-->
36   <!--ELEMENT Reference(DigestMethod, DigestValue)-->
37   <!--ATTLIST Reference URI CDATA #REQUIRED-->
38   <!--ELEMENT DigestMethod(#PCDATA)-->
39   <!--ATTLIST DigestMethod Algorithm CDATA #REQUIRED-->
40   <!--ELEMENT DigestValue(#PCDATA)-->
41   <!--ELEMENT SignatureValue(#PCDATA)-->
42 ]>

```



【図 14】

図 14

```

01<?xml version="1.0"?>
02<!DOCTYPE PurchaseOrder SYSTEM "PurchaseOrder.dtd">
03<PurchaseOrder Id="po">
04   <UserID>10194970</UserID>
05   <Price>100000</Price>
06   <EncryptedData Id="poED">
07     <EncryptionMethod Algorithm="http://www.w3.org/xmlenc#aes128"/>
08     <KeyInfo>
09       <RetrievalMethod Type="http://www.w3.org/xmlenc#EncryptedKey"
10         URI="#poEK"/>
11     </KeyInfo>
12     <CipherData>
13       <CipherValue>SrYKz0a6iu/gi.....y5UZhTTaY9</CipherValue>
14     </CipherData>
15   </EncryptedData>
16   <EncryptedKey Id="poEK">
17     <EncryptionMethod Algorithm="http://www.w3.org/xmlenc#rsa"/>
18     <KeyInfo>
19       <KeyName>poWrapKey</KeyName>
20     </KeyInfo>
21     <CipherData>
22       <CipherValue>kjZVmJbShov4v.....wqbYwQri7QH</CipherValue>
23     </CipherData>
24     <ReferenceList>
25       <DataReference URI="#poED"/>
26     </ReferenceList>
27   </EncryptedKey>
28   <Signature>
29     <SignedInfo>
30       <CanonicalizationMethod Algorithm="http://www.w3.org/xml#canonical"/>
31       <SignatureMethod Algorithm="http://www.w3.org/xmldsig#rsa-sha1"/>
32       <Reference URI="#po">
33         <DigestMethod Algorithm="http://www.w3.org/xmldsig#sha1"/>
34         <DigestValue>AZAOVqTorSSJ70BCA/tLY93rFM=</DigestValue>
35       </Reference>
36     </SignedInfo>
37     <SignatureValue>ZknUOaJsxNR5.....1nHhIG25PKg==</SignatureValue>
38     <KeyInfo>
39       <KeyName>Hitachi.SDL</KeyName>
40     </KeyInfo>
41   </Signature>
42</PurchaseOrder>

```

1401

1411

1412

1413

【図 1 5】

図 1 5

#	種別	適用場所	操作要素	関連文書構造定義
4	Add	Signature.dtd: /Signature/last()	PurchaseOrder.dtd: /PurchaseOrder	PurchaseOrder.dtd KeyInfo.dtd

1511

【図 16】

図 16

1611

```

01<!DOCTYPE Signature [
02  <!ELEMENT Signature(SignedInfo,SignatureValue,KeyInfo?,PurchaseOrder:*)>
03  <!ELEMENT SignedInfo(CanonicalizationMethod,SignatureMethod,Reference+)>
04  <!ELEMENT CanonicalizationMethod(#PCDATA)>
05  <!--ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED-->
06  <!ELEMENT SignatureMethod(#PCDATA)>
07  <!--ATTLIST SignatureMethod Algorithm CDATA #REQUIRED-->
08  <!ELEMENT Reference(DigestMethod,DigestValue)>
09  <!--ATTLIST Reference URI CDATA #REQUIRED-->
10  <!ELEMENT DigestMethod(#PCDATA)>
11  <!--ATTLIST DigestMethod Algorithm CDATA #REQUIRED-->
12  <!ELEMENT DigestValue(#PCDATA)>
13  <!ELEMENT SignatureValue(#PCDATA)>
14  <!--ELEMENT PurchaseOrder(UserID,Price,CreditCard)
15    <!--ATTLIST PurchaseOrder Id ID #IMPLIED-->
16  <!--ELEMENT UserID(#PCDATA)>
17  <!--ELEMENT Price(#PCDATA)>
18  <!--ELEMENT CreditCard(Issuer,Number,Expire,Owner)>
19  <!--ELEMENT Issuer(#PCDATA)>
20  <!--ELEMENT Number(#PCDATA)>
21  <!--ELEMENT Expire(#PCDATA)>
22  <!--ELEMENT Owner(#PCDATA)>
23  <!--ELEMENT KeyInfo(EncryptedKey?,(RetrievalMethod | KeyName))>
24  <!--ELEMENT RetrievalMethod(#PCDATA)>
25  <!--ATTLIST RetrievalMethod
26    Type          CDATA          #REQUIRED
27    URI           CDATA          #REQUIRED
28  <!--ELEMENT KeyName(#PCDATA)>
29]>

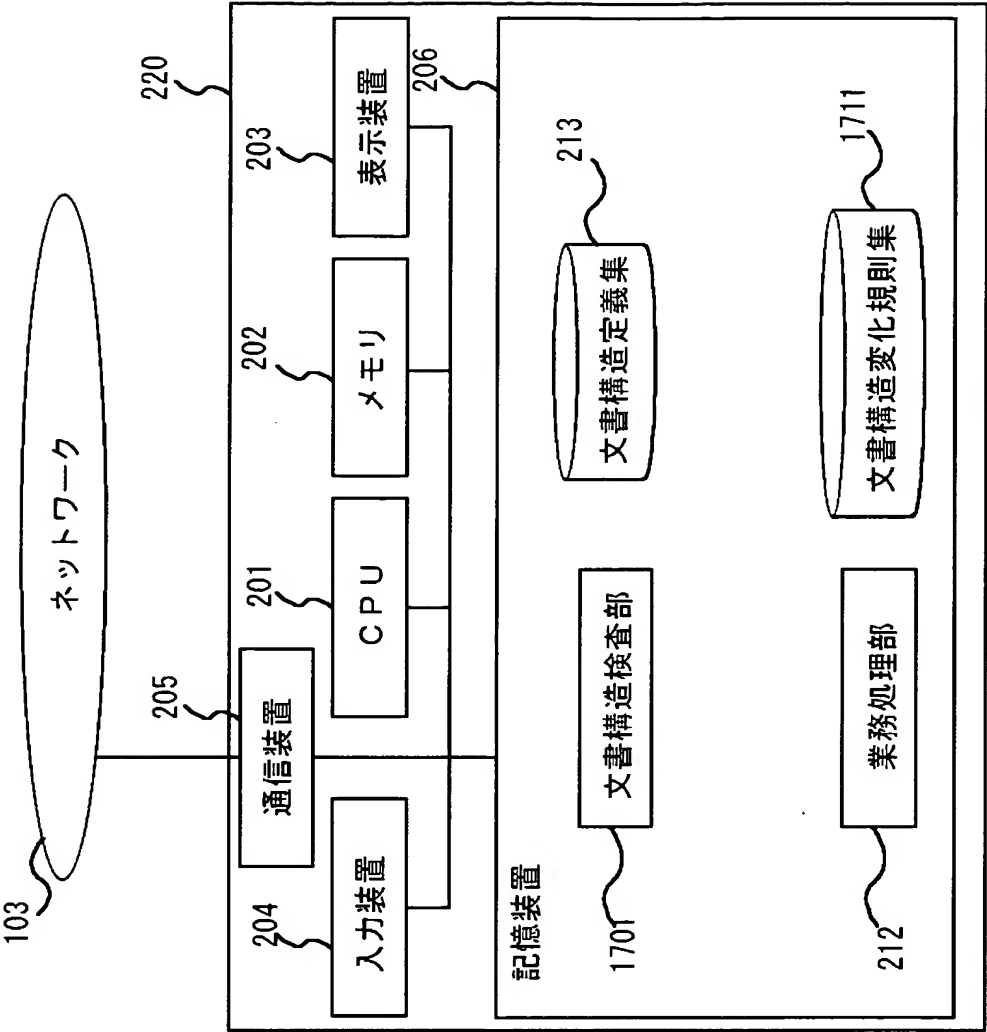
```

1612

1613

【図 17】

図 17



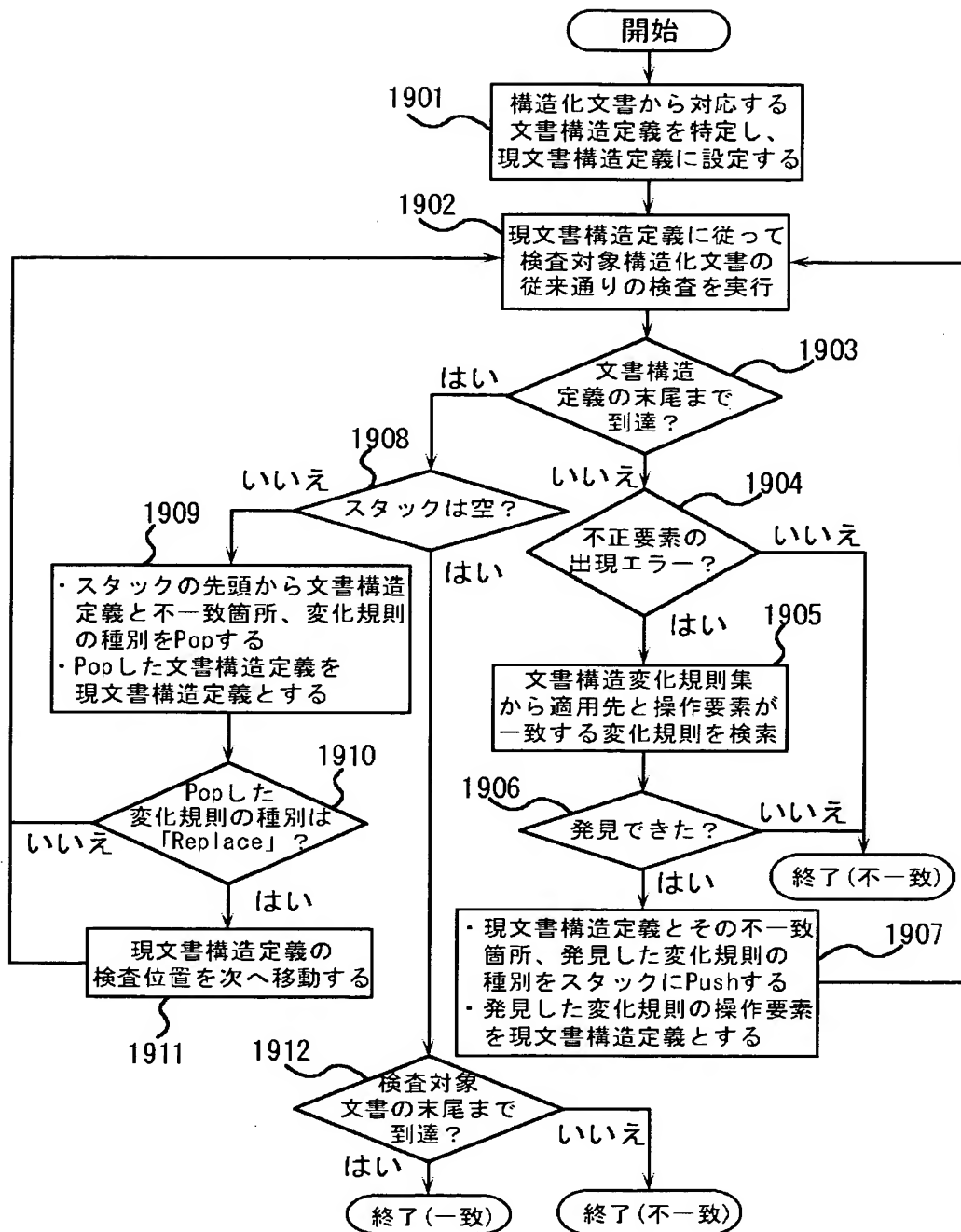
【図 1 8】

図 1 8

1805 {	1801 {	1802 {	1803 {	1804 {
#	種別	適用場所	操作要素	関連文書構造定義
1811 {	1 Replace	*	EncryptedData.dtd: /EncryptedData	EncryptedData.dtd KeyInfo.dtd
1812 {	2 Add	*	EncryptedKey.dtd: /EncryptedKey	EncryptedKey.dtd KeyInfo.dtd
1813 {	3 Add	*	Signature.dtd: /Signature	Signature.dtd KeyInfo.dtd
1814 {	4 Add	Signature.dtd	PurchaseOrder.dtd: /PurchaseOrder	PurchaseOrder.dtd: KeyInfo.dtd

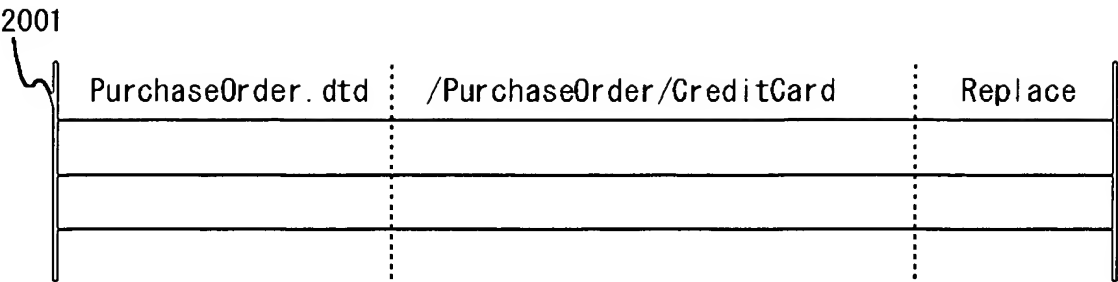
【図 19】

図 19



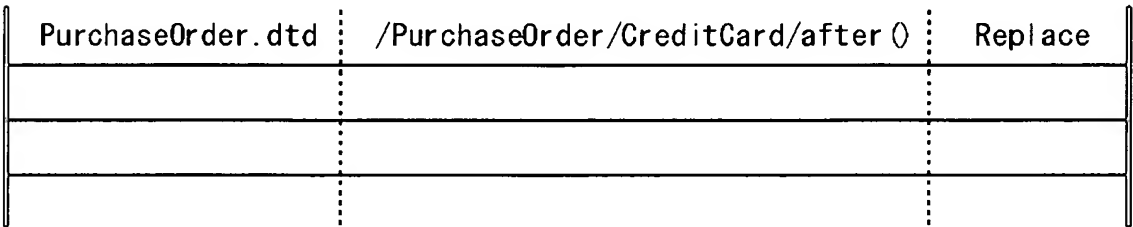
【図 2 0】

図 2 0



【図 2 1】

図 2 1



【図 22】

図 22

2201

```

01<!DOCTYPE PurchaseOrder [
02 <!ELEMENT PurchaseOrder (UserID, Price, EncryptedData)>
03 <!--ATLST PurchaseOrder Id ID #IMPLIED-->
04 <!ELEMENT UserID(#PCDATA)>
05 <!ELEMENT Price(#PCDATA)>
06 <!ELEMENT CreditCard(Issuer, Number, Expire, Owner)>
07 <!ELEMENT Issur(#PCDATA)>
08 <!ELEMENT Number(#PCDATA)>
09 <!ELEMENT Expire(#PCDATA)>
10 <!ELEMENT Owner(#PCDATA)>
11 <!--ENTITY %EncryptedDataRef SYSTEM "EncryptedData.dtd"-->
12 %EncryptedDataRef;
13 <!--ENTITY % KeyInfoRef SYSTEM "KeyInfo.dtd"-->
14 %KeyInfoRef;
15 ]>

```

2211

2212



【書類名】 要約書

【要約】

【課題】

構造化文書に署名付与、暗号化などの処理を施すと文書の構造が変化するため、署名付与、暗号化などの処理を施さない文書と同じ方法で構造検査を実施することができない。

【解決手段】

文書に対する署名付与の場所と方式と署名データの構造、暗号化対象のデータと暗号化方式など文書構造を変化させる処理の内容をすべて事前に取り決め、文書構造検査時に利用される文書構造定義を検査前に署名付与／暗号化後の形式に変換し、該変換後の文書構造定義を利用して文書構造検査を実施する。

本発明により、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、暗号化部分の検査者の復号化可否に関わらず、文書全体の構造検査を実施することが可能となる。

【選択図】 図 2

特願 2 0 0 3 - 1 4 0 5 6 8

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 1 0 8 ]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台 4 丁目 6 番地
氏 名	株式会社日立製作所